



Friends

Link submit:

https://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&page=show_problem&problem=1549

Solution:

C++	https://ideone.com/KKCvAj
Java	https://ideone.com/rkeS1t
Python	https://ideone.com/V7qQlz

Tóm tắt đề:

Ta có N người, và ta biết rằng sẽ có một số cặp người nào đó là bạn của nhau. Và ở đây có một tính chất rằng những người bạn của bạn mình sẽ chính là bạn của mình. Như vậy những người là bạn với nhau sẽ tạo thành 1 nhóm. Đề yêu cầu bạn hãy đếm xem nhóm có số lượng người lớn nhất là bao nhiêu.

Input

Dòng đầu gồm một số nguyên dương T là số lượng bộ test

T bộ test sau, mỗi bộ test được tổ chức theo dạng như sau:

- Dòng 1 gồm 2 số nguyên dương N và M , tương ứng là số lượng người và M cặp quan hệ bạn bè giữa họ. ($1 \leq N \leq 30,000$, $1 \leq M \leq 500,000$).
- M dòng sau, mỗi dòng sẽ chứa 2 số nguyên dương A , B thể hiện rằng người A và B là bạn với nhau.

Output

In ra T dòng ứng với T bộ test, dòng thứ i là số lượng người trong group lớn nhất ở test thứ i .

2 3 2 1 2 2 3 10 12 1 2 3 1	3 7
---	--------

3 4	
5 4	
3 5	
4 6	
5 2	
2 1	
7 1	
1 2	
9 10	
8 9	

Giải thích: Trong ví dụ đầu có 3 người sau khi kết nối người 1 với người 2 lại, người 2 với người 3 lại thì chúng ta có một nhóm số lượng người lớn nhất là 3 người.

Hướng dẫn giải:

Ta ứng dụng DSU, nhưng thay vì với DSU cơ bản, ta chỉ có `parent[u]` để lưu lại cha của `u` trong tập hợp chứa `u`, thì lúc này ta thêm 1 mảng nữa là `cnt` với `cnt[u]` là số lượng phần tử trong tập nhận `u` làm gốc. Khi ta gộp 2 tập hợp lại với nhau, thì số lượng phần tử sẽ tăng lên, giả sử nếu ta gộp tập hợp gốc `v` sang tập hợp gốc `u`, thì ngoài `parent[v] = u` thì ta sẽ cập nhật thêm `cnt[u] += cnt[v]`.

Sau khi duyệt hết các cạnh và thực hiện union các cặp đỉnh, ta thu được một mảng `cnt`, chỉ cần tìm phần tử lớn nhất trong mảng là được (tức nhóm bạn đông nhất).

Độ phức tạp: $O(T \cdot N \cdot M)$ trong trường hợp sử dụng DSU cơ bản, nếu sử dụng DSU nâng cao (update by rank, path compression, by size) thì sẽ là $O(T \cdot M \cdot \log(N))$